CLAIMS

What is claimed is:

1. An odometer counter, said odometer counter comprising:

an encoder, said encoder having a unique key, said key constructed and arranged to configure an encryption engine, said encoder further constructed and arranged to execute said encryption engine with an odometer value to form an encrypted odometer value, said encoder further constructed and arranged to store said odometer value in a non-volatile memory; and

a decoder, said decoder constructed and arranged to receive said encrypted odometer value from said encoder, said decoder further constructed and arranged to decrypt said encrypted odometer value and to transmit said decrypted odometer value.

2. An odometer counter as in claim 1 wherein said encoder is further constructed and arranged to increment said odometer value upon receiving an increment signal.

3. An odometer counter as in claim 1 wherein said key is 64 bits in length.

4. An odometer counter as in claim 1 wherein odometer value is 16 bits in length.

5. An odometer counter as in claim 4 wherein said 16 bit odometer value is combined with another 16 bit value.

6. An odometer counter as in claim 5 wherein said another 16 bit value is composed of

a pre-defined 12 bit value and a 4 bit value based upon inputs to said encoder.

7. An odometer counter as in claim 6 wherein said 12 bit value is stored in said non-volatile memory of said encoder.

8. An odometer counter as in claim 1 wherein said data packet from said encoder is transmitted asynchronously to said decoder.

9. An method for securing odometer values, said method comprising the steps of:

(a) providing an encoder having a unique key and an encryption engine;

(b) configuring said encryption circuit with said key;

(c) receiving an increment signal;

(d) incrementing an odometer value to form an incremented odometer value;

(e) storing said incremented odometer value into said a non-volatile memory;

(e) encrypting said incremented odometer value with said encryption engine to form an encrypted odometer value;

(f) transmitting said encrypted odometer value to a decoder; and

(g) decrypting said encrypted odometer value with said decoder to obtain said odometer value.

10. A method as in claim 9, wherein said step of encrypting said odometer value with said encryption engine to form an encrypted odometer value comprises the following steps:

(e1) adding a 16-bit value to said odometer value to form a 32 bit value; and

(e2) encrypting said 32 bit value to form said encrypted odometer value.

11. A method as in claim 9, wherein said step of transmitting said encrypted odometer value to a decoder comprises the following steps:

(f1) generating a data packet with said encrypted odometer value;

(f2) adding an identifier code to said data packet to form a tagged data packet; and.

(f3) transmitting said tagged data packet from said encoder to said decoder.

12. An odometer counter as in claim 6 wherein a portion of said 12 bit value is stored in said non-volatile memory of said encoder.

13. An odometer counter as in claim 6 wherein a portion of said 12 bit value is a checksum based on said odometer value.

14. A counter, said counter comprising:

an encoder, said encoder having a unique key, said key constructed and arranged to configure an encryption engine, said encoder further constructed and arranged to execute said encryption engine with an value to form an encrypted value, said encoder further constructed and arranged to store said value in a non-volatile memory; and

a decoder, said decoder constructed and arranged to receive said encrypted value from said encoder, said decoder further constructed and arranged to decrypt said encrypted value and to transmit said decrypted value.

15. A counter as in claim 14 wherein said encoder is further constructed and arranged to increment said value upon receiving an increment signal.

16. A counter as in claim 14 wherein said key is 64 bits in length.

17. A counter as in claim 14 wherein value is 16 bits in length.

18. A counter as in claim 17 wherein said 16 bit value is combined with another 16 bit value.

19. A counter as in claim 18 wherein said another 16 bit value is composed of a pre-defined 12 bit value and a 4 bit value based upon inputs to said encoder.

20. A counter as in claim 19 wherein said 12 bit value is stored in said non-volatile memory of said encoder.

21. A counter as in claim 14 wherein said data packet from said encoder is transmitted asynchronously to said decoder.

22. An method for securing values, said method comprising the steps of:

(a) providing an encoder having a unique key and an encryption engine;

(b) configuring said encryption circuit with said key;

(c) receiving an increment signal;

(d) incrementing an value to form an incremented value;

(e) storing said incremented value into said a non-volatile memory;

(e) encrypting said incremented value with said encryption engine to form an encrypted value;

(f) transmitting said encrypted value to a decoder; and

(g) decrypting said encrypted value with said decoder to obtain said value.

23. A method as in claim 22, wherein said step of encrypting said value with said encryption engine to form an encrypted value comprises the following steps:

(e1) adding a 16-bit value to said value to form a 32 bit value; and

(e2) encrypting said 32 bit value to form said encrypted value.

24. A method as in claim 22, wherein said step of transmitting said encrypted value to a decoder comprises the following steps:

(f1) generating a data packet with said encrypted value;

(f2) adding an identifier code to said data packet to form a tagged data packet; and.

(f3) transmitting said tagged data packet from said encoder to said decoder.

25. A counter as in claim 6 wherein a portion of said 12 bit value is stored in said non-volatile memory of said encoder.

26. A counter as in claim 19 wherein a portion of said 12 bit value is a checksum based on said value.